CLAIMS

What is claimed is:

1. A computer implemented method for establishing a run-time data area comprising:

relocating a firmware module from a read-only memory location to a writeable memory location during a system boot-up operation;

reserving a portion of said writeable memory location comprising a memory allocation for said firmware module and an additional memory allocation; and

designating said additional memory allocation as said run-time data area, wherein said run-time data area is created without requiring prior knowledge of system resource allocation.

2. The computer implemented method as recited in Claim 1 wherein said relocating further comprises:

receiving a system call for a system firmware feature; and

returning a response to said system call requesting said memory allocation for said firmware module, said additional memory allocation, and a memory allocation for said system firmware feature.

3. The computer implemented method as recited in Claim 2 further comprising:

determining the size of said system firmware feature;

determining the size of said firmware module; and

determining the size of said run-time data area.

4. The computer implemented method as recited in Claim 2 wherein said system firmware feature comprises a processor abstraction layer.

5. The computer implemented method as recited in Claim 1 wherein said firmware module operates in a real mode.

6. The computer implemented method as recited in Claim 1 wherein said firmware module operates in a virtual mode.

7. The computer implemented method as recited in Claim 1 wherein said firmware module is dynamically operable in a real mode and a virtual mode.

8. A method for creating a system independent run-time data storage area comprising:

intercepting a system call for determining the size of a system firmware feature during a system boot-up operation;

returning a response to said system call conveying a request for a portion of a writeable memory location; and

reserving a portion of said writeable memory location, wherein a memory allocation is designated as said run-time data area, wherein said run-time data area is created without requiring prior knowledge of system resource allocation.

9. The method as recited in Claim 8 further comprising:

utilizing a firmware module resident upon a read-only memory location to perform said intercepting.

10. The method as recited in Claim 9 further comprising:

relocating said system firmware feature and said firmware module from said read-only memory location to said writeable memory location.

11. The method as recited in Claim 10 wherein said run-time data area comprises a sub-component of said firmware module.

12. The method as recited in Claim 10 wherein said run-time data area is separate from said firmware module and said system firmware feature.

13. The method as recited in Claim 8 wherein said system boot-up operation is performed by an Intel Itanium® processor.

14. The method as recited in Claim 13 wherein said system firmware feature comprises a processor abstraction layer.

15. The as recited in Claim 9 wherein said firmware module operates in a real mode.

16. The method as recited in Claim 9 wherein said firmware module operates in a virtual mode.

17. The method as recited in Claim 9 wherein said firmware module is dynamically operable in a real mode and a virtual mode.

18. A method for creating a run-time data area comprising:

receiving a system call for relocating a system firmware feature from a read-only memory location to a writeable memory location during a system boot-up operation;

allocating a first portion of said writeable memory location for said system firmware feature; and

allocating an additional portion of said writeable memory location and designating said additional memory allocation as said run-time data area, wherein said run-time data area is created without requiring prior knowledge of system resource allocation.

19. The method as recited in Claim 18 wherein said system firmware feature comprises a processor abstraction layer.

20. The method as recited in Claim 18 further comprising:

using a firmware module to perform said intercepting.

21. The method as recited in Claim 20 further comprising:

allocating a third portion of said writeable memory location to said firmware module.

22. The method as recited in Claim 20 further comprising:

allocating said additional portion of said writeable memory location to said firmware module; and

designating a portion of said firmware module as said run-time data area.

23. The method as recited in Claim 20 wherein said firmware module operates in a real mode.

24. The computer implemented method as recited in Claim 20 wherein said firmware module operates in a virtual mode.

25. The computer implemented method as recited in Claim 20 wherein said firmware module is dynamically operable in a real mode and a virtual mode.